Optimization–Based Control of Constrained Nonlinear Systems with Continuous–Time Models: Adaptive Time–Grid Refinement Algorithms

Fernando A.C.C. Fontes[1,a)] and Luís T. Paiva[1,b)]

[1]*Systec–ISR, Faculdade de Engenharia, Universidade do Porto*
4200-465 Porto, Portugal

[a)]Corresponding author: faf@fe.up.pt
[b)]ltpaiva@fe.up.pt

Abstract. We address optimal control problems for nonlinear systems with pathwise state-constraints. These are challenging nonlinear problems for which the number of discretization points is a major factor determining the computational time. Also, the location of these points has a major impact in the accuracy of the solutions. We propose an algorithm that iteratively finds an adequate time-grid to satisfy some predefined error estimate on the obtained trajectories, which is guided by information on the adjoint multipliers. The obtained results show a highly favorable comparison against the traditional equidistant–spaced time–grid methods, including the ones using discrete–time models. This way, continuous–time plant models can be directly used. The discretization procedure can be automated and there is no need to select a priori the adequate time step. Even if the optimization procedure is forced to stop in an early stage, as might be the case in real–time problems, we can still obtain a meaningful solution, although it might be a less accurate one. The extension of the procedure to a Model Predictive Control (MPC) context is proposed here. By defining a time–dependent accuracy threshold, we can generate solutions that are more accurate in the initial parts of the receding horizon, which are the most relevant for MPC.

INTRODUCTION

In this paper, we discuss an adaptive time–mesh refinement (AMR) algorithm to efficiently and accurately solve optimal control problems (OCP) and we propose its use in Model Predictive Control (MPC) schemes by extending the AMR to accommodate a time–varying accuracy threshold for the maximum error estimates.

Nowadays, the numerical solution of nonlinear optimal control problems is most frequently done using the so–called direct methods. Regular time meshes having equidistant spacing are also most frequently used. However, in some cases these meshes cannot accurately cope with nonlinear behavior. One way to improve the solution is to select a new mesh with a greater number of nodes. Another way, used here, involves adaptive mesh refinement. In this case, the mesh nodes have non equidistant spacing which allow a non uniform nodes collocation – adding more nodes where they are more needed. In real–time optimization, *e.g.* Model Predictive Control (MPC), where a solution must be produced within a time–frame, AMR has the additional advantage that lower accuracy solutions are produced at a very early stage and then the method can improve the accuracy of the solutions within the time available. As such, with AMR, even when the real–time optimization is interrupted at an early stage, a solution is available.

Obtaining an adapted mesh through iterative refinement is a widely studied area, for example in the context of partial differential equations. In optimal control there has been some contributions, see [1, 2, 3] and references therein. Here, we follow closely the work [4], for which a preliminary version appeared in [5]. The main differences to the previous works are the fact that ours uses information of the multipliers to guide the refinement, as well as different levels of refinement in a single iteration. The main contribution is the adaptation of the AMR for MPC which requires the extension to time–varying notions in the problem setting as well as in the procedures.

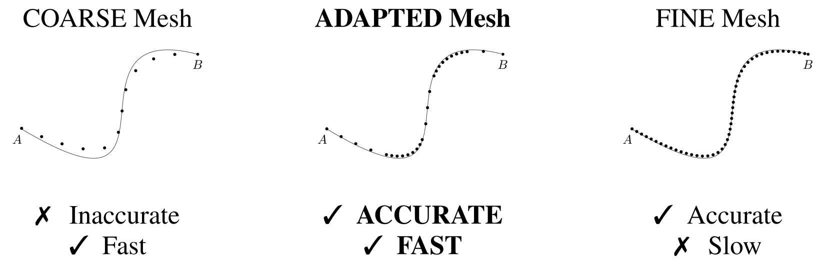


FIGURE 1: Adapted time meshes enable accurate and fast solutions

OPTIMAL CONTROL AND THE AMR ALGORITHM

We consider the following nonlinear optimal control problem with pathwise state constraints:

$$\mathcal{P}(t_0, t_f) : \text{Minimise} \quad \int_{t_0}^{t_f} \mathrm{L}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) \mathrm{d}t + \mathrm{G}\left(\mathbf{x}(t_f)\right) \tag{1}$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \qquad \text{a.e. } t \in \left[t_0, t_f\right], \tag{2}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0\,, \quad \mathbf{x}(t_f) \in \mathbb{X}_1 \subset \mathbb{R}^n\,, \tag{3}$$

$$\mathbf{x}(t) \in \mathbb{X} \subset \mathbb{R}^n \qquad \forall t \in \left[t_0, t_f\right], \tag{4}$$

$$\mathbf{u}(t) \in \mathbb{U} \subset \mathbb{R}^m \qquad \text{a.e. } t \in \left[t_0, t_f\right], \tag{5}$$

The state constraint (4) is typically implemented as an inequality constraint (or set of inequality constraints) $h(x(t)) \leq 0$, $\forall t \in \left[t_0, t_f\right]$, where the function h can be obtained from the set inclusion using the distance or the oriented distance functions to a set (cf.[6, 7]).

In order to numerically solve this problem using direct methods, we need to select a time grid $\pi := \{t_i\}_{i\geq 0}$ in $[t_0, t_f]$ where $t_{i+1} = t_i + \delta_i$ and $\delta_i > 0$. The dynamics constraint $\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$, a.e. $t \in [t_0, t_f]$ has a discrete approximation in π, *e.g.* , $\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) + \delta_i \mathbf{f}(t_i, \mathbf{x}(t_i), \mathbf{u}(t_i))\,, t_i \in \pi$ if we consider the 1$^{\text{st}}$ Euler scheme.

As is well known there is an error associated with this or any approximations. Let ϵ_π be an estimate for the discretization error. The discretization procedures are always based on some form of local linearization, though they might have higher order terms. So, we should expect non–negligible errors for highly nonlinear systems.

When selecting a mesh using Direct Methods, the number of discretization points is a major factor regarding the computational time. Also, the location of these points has a major impact in the accuracy of the solutions. To have a fast and accurate solution we need to place the nodes where they are most needed, which can only be achieved with an iteratively adapted grid (see Fig. 1).

To solve numerically optimal control problems, we follow closely the work of [4]. The adaptive mesh refinement process starts by discretizing the time interval $[t_0, t_f]$ in a coarse mesh in order to acquire the main structure of the solution. According to some refinement criteria, the mesh is divided in K mesh intervals $\mathcal{S}_k = [\tau_{k-1}, \tau_k[\,, k = 1, \ldots, K$, where $(\tau_0, \ldots, \tau_K)$ coincide with nodes. These mesh intervals $\mathcal{S}_k$ form a partition of the time interval while the mesh nodes have the property $\tau_0 < \tau_1 < \ldots < \tau_K$. The subintervals $\mathcal{S}_k$ that verify the refinement criteria are refined taking into account different levels of refinement in a single iteration, *i.e.* , they are divided into smaller subintervals according to user–defined levels of refinement $\bar{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m]$. The procedure is repeated until the stopping criterion is achieved. A subinterval $\mathcal{S}_{k,j}$ is at the j^{th} level of refinement if

$$S_{k,j} = \left\{t \in S_k \;:\; \varepsilon(t) \in \left[\varepsilon_j, \varepsilon_{j+1}\right[\right\} \tag{6}$$

for $j = 1, \ldots, m$. This procedure adds more node points to the subintervals in higher levels of refinement, corresponding to higher errors, and it adds less node points to those in lower refinement levels. By defining several levels of refinement, we get a multi–level time–mesh in a single iteration.

We need to define a refinement criteria and a stopping criterion used. As refinement criteria we use the estimate of the relative error of the adjoint multipliers (dual variables). For the stopping criterion, we consider a threshold for the relative error estimate of the trajectory. In the refinement test, we consider the multipliers $\mathbf{q}_{\text{MP}}$ which are solution of the adjoint differential equation system in the Maximum Principle [6, 7], and we also consider the multipliers $\mathbf{q}_{\text{KKT}}$ obtained by applying the Kuhn–Tucker conditions to nonlinear optimization problem which results from the transcription of the optimal control. The relative error estimate is, at each time, the difference between the multipliers $\mathbf{q}_{\text{KKT}}$ computed by the numerical solver and $\mathbf{q}_{\text{MP}}$ computed by integrating numerically the adjoint equation given by the Maximum Principle. In addition, $\mathbf{q}_{\text{MP}}$ are a solution to a linear differential equation system, which can be easily solved in a faster way and with higher accuracy. As stopping criterion, we consider the L^∞ norm of the relative error of the primal variables ($\varepsilon_{\mathbf{x}}$). Since the proposed procedure increases the number of nodes, more computational time would be expected. To decrease the CPU time, when going from a coarse mesh to a refined one, the previous solution is used as warm start for the next iteration. To create this warm start, the solution obtained in the coarse mesh is projected in the refined one using the cubic Hermite interpolation. This action proved to be vital in the decreasing of the overall computational time.

The numerical results in [4] show that, using the AMR algorithm, the OCP can be solved considerably faster (up to 38 times faster in some applications) for the same level of accuracy than using an equidistant time-grid, as is done when discrete–time models are used.

THE AMR ALGORITHM IN MPC

The MPC technique is a procedure used to generate control laws dependent on the current (measured) state of the plant by solving on–line a sequence of finite horizon open–loop OCP subject to system dynamics and constraints involving states and controls. Based on measurements obtained at the time instant t_i, the controller predicts the future input such that a predetermined open–loop performance objective functional is optimized. Then, the open–loop control is implemented until the next measurement becomes available. Using the new measurement at the time instant $t_i + \delta_k$, where δ_k is the sampling time step, the whole procedure – prediction and optimization – is repeated to find a new input function with the control and prediction horizon moving forward [8, 9]. The sampling step of the MPC procedure is often considered to be fixed, *i.e.* , the measurement takes place every $\delta_k = \delta$ sampling time–units.

For numerical solutions of the open–loop optimal control problem, it is often necessary to parameterize the input in an appropriate way. This is normally done by using a finite number of basis functions, *e.g.* , the input could be approximated as piecewise constant over the sampling time δ.

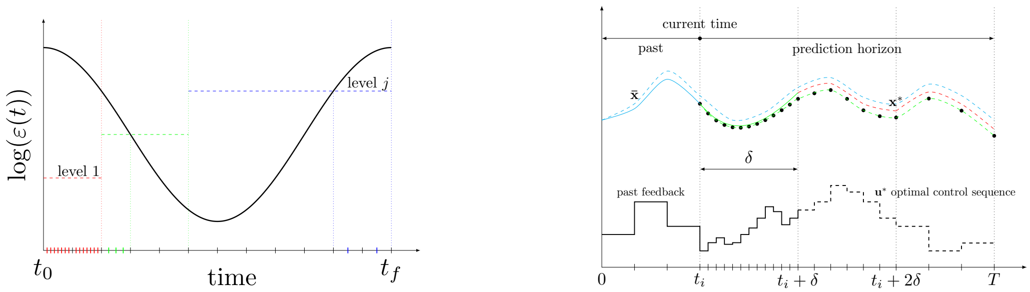
Let us consider a sampling step $\delta > 0$, the prediction horizon T and a sequence of sampling instants $\{t_i\}_{i\geq 0}$ with $t_{i+1} = t_i + \delta$. The sampled–data MPC algorithm follows the receding horizon strategy:

1. Measure state of the plant $\mathbf{x}_{t_i}$;
2. Determine $\bar{\mathbf{u}} : [t_i, t_i + T] \to \mathbb{R}^m$ solution to the OCP $\mathcal{P}(t_i, t_i + T)$: (1)-(6).
3. Apply the control $\mathbf{u}^*(t) := \bar{\mathbf{u}}(t)$ to the plant in the interval $t \in [t_i, t_i + \delta]$, disregarding the remaining control $\bar{\mathbf{u}}(t), t > t_i + \delta$;
4. Repeat this procedure for the next sampling time instant $t_i + \delta$.

We extend the adaptive time–mesh refinement algorithm described in [4] in order to allow different refinement levels according to some partition of the time domain. This extension is of relevance in the MPC context, since it is desirable to have a solution with higher accuracy in the initial part of the horizon.

The time interval $t \in \left[t_0, t_f\right]$, the prediction horizon T, and the sampling step $\delta > 0$ satisfy $\delta << T << t_f - t_0$. When applying the MPC procedure to solve an OCP, at each time instant t_i we compute the solution in $[t_i, t_i + T]$ but we just implement the open–loop control until $t_i + \delta$. Therefore, taking into account the planning strategy discussed above, it would be an improvement if we have an adaptive time–mesh able to cope this feature, *i.e.* , a time–mesh that is highly refined in the lower limit of the time interval $[t_i, t_i + T]$ and it is coarser in the upper limit of the same interval. Then, we would implement the control on the time interval $[t_i, t_i + \delta]$ computed with high accuracy in a refined mesh. For the remaining time interval we have an estimate of the solution.

In this extension, we also consider a time–dependent stopping criterion for the mesh refinement algorithm with different levels $\bar{\varepsilon}_{\mathbf{x}}(t)$. Instead of having a fixed stopping criterion $\varepsilon_{\mathbf{x}}^{\max}$, now we have a time–dependent $\bar{\varepsilon}_{\mathbf{x}}(t)$ stopping criterion which sets different levels of accuracy for the solution, along the time domain. For example, the time–dependent levels of refinement can be defined as $\bar{\varepsilon}_{\mathbf{x}}(t) = \begin{cases} \varepsilon_{\mathbf{x}}^{\max}, & t \in [t_i, t_i + \beta_1 T] \\ \alpha_1 \varepsilon_{\mathbf{x}}^{\max}, & t \in \left]t_i + \beta_1 T, t_i + \beta_2 T\right] \\ \dots & \\ \alpha_j \varepsilon_{\mathbf{x}}^{\max}, & t \in \left]t_i + \beta_j T, t_i + T\right] \end{cases}$, where $1 < \alpha_1 < \ldots <$



(a) Time–dependent thresholds $\bar{\varepsilon}_{\mathbf{x}}(t)$ (b) Time–mesh refinement along the MPC horizon

FIGURE 2: The extended (time–dependent) refinement algorithm for MPC

$\alpha_j \leq \varepsilon_{\mathbf{x}}^{\max}$ and $0 < \beta_1 < \beta_2 < \ldots < \beta_j < 1$.

This procedure adds more node points to the subintervals that are in lower levels of the stopping criterion for the refinement procedure, corresponding to time instants close to the initial time as illustrated in Fig. 2(a).

Model Predictive Control coupled with the Extended Algorithm. We start the MPC procedure in the typical way but considering an adaptive mesh refinement strategy. We descritise the time interval $\left[t_0, t_f\right]$ and we solve our OCP in open–loop. Then, we implement the control in the first sampling interval. When starting the next MPC step, we apply the time–mesh refinement strategy in order to find the best mesh suited to the solve the OCP in the second sampling interval (Fig. 2(b)). In the MPC algorithm, step 2 is modified as follows:

2. (a) Select the intervals $S_{k,j}$ to be refined according to the time–dependent levels of refinement $\bar{\varepsilon}_{\mathbf{x}}(t)$ and generate a new time grid;
   (b) Determine $\bar{\mathbf{u}} : [t_i, t_i + T] \to \mathbb{R}^m$ solution to the OCP $\mathcal{P}(t_i, t_i + T) : (1)$-(6), in the new time–grid.

CONCLUDING REMARKS

The refinement algorithm proposed for MPC has a time–dependent stopping criterion ensuring a solution with higher accuracy in the initial part of the horizon. Due to its fast response, the AMR algorithm coupled with MPC can be used to solve fast real–time optimization problems. Further details and examples can be seen in [10].

ACKNOWLEDGMENTS

We acknowledge the support of FEDER/COMPETE2020-POCI/FCT funds through POCI-01-0145-FEDER-006933-SYSTEC and PTDC/EEI-AUT/2933/2014.

REFERENCES

[1] J. T. Betts and W. P. Huffman, Optimal Control Applications and Methods **19**, 1–21 (1998).
[2] Y. Zhao and P. Tsiotras, Journal of Guidance, Control, and Dynamics **34**, 271–277 (2011).
[3] M. A. Patterson, W. W. Hager, and A. V. Rao, Optimal Control Applications and Methods (2014).
[4] L. T. Paiva and F. A. Fontes, Discrete and Continuous Dynamical Systems **35**, 4553–4572 (2015).
[5] L. T. Paiva and F. A. C. C. Fontes, "Mesh refinement strategies for optimal control problems," in *11th International Conference on Numerical Analysis and Applied Mathematics*, AIP Proceedings (AIP, 2013).
[6] F. A. Fontes and H. Frankowska, Journal of Optimization Theory and Applications **166**, 115–136 (2015).
[7] R. B. Vinter, *Optimal Control* (Springer, 2000).
[8] F. A. C. C. Fontes, Systems and Control Letters **42**, 127–143 (2001).
[9] F. A. C. C. Fontes, L. Magni, and E. Gyurkovics, in *Assessment and Future Directions of Nonlinear Model Predictive Control*, edited by F. Allgöwer, R. Findeisen, and L. Biegler (Springer Verlag, 2007), pp. 115–129.
[10] L. T. Paiva, "Numerical Methods in Optimal Control and Model Predictive Control," Ph.D., U.Porto 2014.